情報工学実験３レポート

C言語によるアフィン変換の実装

15X3034 音賀　優颯

wasabi

1. 目的

　プログラミング言語により，アフィン変換を実装することで，理解を深める．

1. 内容

　C言語を用いて任意の拡大率，回転角を与えることで，拡大，回転された画像が保存されるプログラムを作成した．また，出力する画像の各画素はバイリニア補間を用いた．

1. コード

以下に作成したコードを示す.

image\_process.c

1 #include <stdio.h>

2 #include <stdlib.h>

3 #include <string.h>

4 #include <math.h>

5

6 typedef struct{

7 char\* magic[3];

8 int row,column,max;

9 unsigned char\*\*\* array;

10 } image\_t;

11

12 int read\_image(char\* input, image\_t\* img)

13 {

14 int i,j,k;

15 char\* buffer;

16 FILE\* fp;

17

18 buffer=calloc(BUFSIZ,sizeof(char\*));

19 sprintf(buffer,"./%s",input);

20 if((fp=fopen(buffer,"rb"))==NULL){

21 fprintf(stderr,"入力ファイルが開けません.\n");

22 exit(1);

23 }

24

25 //read img

26 for(i=0;i<3;i++){

27 fgets(buffer,BUFSIZ,fp);

28 if(i==0)

29 sprintf(img->magic,"%s",(strtok(buffer,"\n")));

30 if(i==1){

31 img->row=atoi(strtok(buffer," "));

32 img->column=atoi(strtok(NULL," "));

33 }

34 if(i==2)

35 img->max=atoi(strtok(buffer," "));

36 }

37 //memoryの動的確保

38 img->array=malloc(3\*sizeof(unsigned char\*\*));

39 for(i=0;i<3;i++){

40 img->array[i]=malloc(img->column\*sizeof(unsigned char\*));

41 for(j=0;j<(img->column);j++){

42 img->array[i][j]=malloc(img->row\*sizeof(unsigned char));

43 }

44 }

45 //読み込み

46 for(j=0;j<(img->column);j++){

47 for(k=0;k<(img->row);k++){

48 fread(&(img->array[0][j][k]),sizeof(unsigned char),1,fp);

49 fread(&(img->array[1][j][k]),sizeof(unsigned char),1,fp);

50 fread(&(img->array[2][j][k]),sizeof(unsigned char),1,fp);

51 }

52 }

53

54 return 0;

55

56 }//end of read\_image

57

58 int affine(double scaler, double theta, image\_t\* img\_in, image\_t\* img\_out)

59 {

60 int i,j,k;//for

61 double sn,cs;//sin,cos

62 sn=sin((theta/180)\*M\_PI);

63 cs=cos((theta/180)\*M\_PI);

64

65 int in\_row,in\_column;//入力画像のサイズ

66 in\_row=img\_in->row;

67 in\_column=img\_in->column;

68

69 int out\_row,out\_column;//出力画像のサイズ

70 double ix,iy;//出力画像の画素から計算した元画像の座標

71 int nx,ny;//出力画像の画素から計算した元画像の座標

72

73 //出力画像のサイズを計算

74 if((theta>=0 && theta<=90) || (theta>=180 && theta<=270)){

75 out\_row=abs(in\_row\*cs)+abs(in\_column\*sn);

76 out\_column=abs(in\_column\*cs)+abs(in\_row\*sn);

77 }

78 else{

79 out\_row=abs(in\_column\*cs)+abs(in\_row\*sn);

80 out\_column=abs(in\_row\*cs)+abs(in\_column\*sn);

81 }

82 out\_row\*=scaler;

83 out\_column\*=scaler;

84

85 //メモリの動的確保

86 img\_out->array=malloc(3\*sizeof(unsigned char\*\*));

87 for(i=0;i<3;i++){

88 img\_out->array[i]=malloc(out\_column\*sizeof(unsigned char\*));

89 for(j=0;j<out\_column;j++){

90 img\_out->array[i][j]=malloc(out\_row\*sizeof(unsigned char));

91 }

92 }

93

94 //画像の中心を軸にして回転させるため,画像サイズの半分ずらす

95 for(i=-out\_column/2;i<out\_column/2;i++){

96 for(j=-out\_row/2;j<out\_row/2;j++){

97

98 //回転後の座標を計算

99 //拡大率でわると元画像が内接する長方形のサイズになる

100 ix=(j\*cs-i\*sn)/scaler;

101 iy=(j\*sn+i\*cs)/scaler;

102

103 //画像の中心が元画像の中心と同じになるように元画像の半分戻す

104 ix+=in\_row/2;

105 iy+=in\_column/2;

106

107 //倍精度をint型にキャスト

108 nx=ix;

109 ny=iy;

110

111 //ix,iyはバイリニア補間の重みになる

112 ix=ix-nx;

113 iy=iy-ny;

114

115

116 if(nx>=0 && ny>=0 && nx<in\_row-1 && ny<in\_column-1){

117 for(k=0;k<3;k++){

118 img\_out->array[k][i+out\_column/2][j+out\_row/2]=(int)(\

119 (double)img\_in->array[k][ny][nx]\*(1-ix)\*(1-iy)+\

120 (double)img\_in->array[k][ny][nx+1]\*ix\*(1-iy)+\

121 (double)img\_in->array[k][ny+1][nx]\*(1-ix)\*iy+\

122 (double)img\_in->array[k][ny+1][nx+1]\*ix\*iy);

123 }

124 }

125 //元画像が値を持っていない（元画像の外側を参照している）

126 else{

127 img\_out->array[0][i+out\_column/2][j+out\_row/2]=150;

128 img\_out->array[1][i+out\_column/2][j+out\_row/2]=150;

129 img\_out->array[2][i+out\_column/2][j+out\_row/2]=150;

130 }

131 }

132 }

133

134 //header部分

135 sprintf(img\_out->magic,"%s",img\_in->magic);

136 img\_out->row=out\_row;

137 img\_out->column=out\_column;

138 img\_out->max=img\_in->max;

139

140 return 0;

141 }//end of affine

142

143 int make\_image(char\* out, image\_t\* img)

144 {

145 int i,j;

146 char\* buffer;

147 FILE\* fp;

148

149 buffer=calloc(BUFSIZ,sizeof(char\*));

150 sprintf(buffer,"./%s",out);

151 if((fp=fopen(buffer,"wb"))==NULL){

152 fprintf(stderr,"出力ファイルが開けません.\n");

153 exit(1);

154 }

155 fprintf(fp,"P6\n");

156 fprintf(fp,"%d %d\n",img->row,img->column);

157 fprintf(fp,"%d\n",img->max);

158 for(i=0;i<img->column;i++){

159 for(j=0;j<img->row;j++){

160 fwrite(&img->array[0][i][j],sizeof(unsigned char),1,fp);

161 fwrite(&img->array[1][i][j],sizeof(unsigned char),1,fp);

162 fwrite(&img->array[2][i][j],sizeof(unsigned char),1,fp);

163 }

164 }

165 return 0;

166

167 }//end of make\_image

168

169 int main(int argc, char\*\* argv)

170 {

171 int i;

172 image\_t img\_in,img\_out;

173

174 if(argc<5){

175 fprintf(stderr,"引数が足りません,入力ファイル \

176 出力ファイル 拡大率 回転角 の順で実行してください.\n");

177 exit(1);

178 }

179

180 read\_image(argv[1],&img\_in);

181 affine(atof(argv[3]),atof(argv[4]),&img\_in,&img\_out);

182 make\_image(argv[2],&img\_out);

183

184 free(img\_in.array);

185 return 0;

186 }//end of main

1. 結果

以下のように実行した

$image\_process lena.ppm modified\_lena.ppm 2.4 30

元の画像をFig. 1に元画像として，Fig. 2に出力画像として示す．





1. 考察
2. 結論